

Syntactic Analysis.

Exercises

Theory of Automata and Discrete Mathematics
School of Computer Science
University of Oviedo

1. New arithmetic operator (average)

Add a new binary operator, `?`, producing the token `AVG`. The syntax is `x?y` where `x` and `y` are arithmetic expressions.

This operation computes $(x + y)/2$. For example, `3?2=2.5`, `2?2=2` and `4?2=3`.

The precedence of this operator is higher than the addition and the subtraction, and lower than the multiplication and the division, forcing left to right evaluation order. Print appropriate messages.

The following program should print 6.5 and 7.5.

```
begin
print(3?2*5);
print(5+3?2);
end
```

2. Booleans

Modify your parser so it admits booleans as boolean expressions. The booleans are `TRUE` and `FALSE` and they will produce the token type `BOOLEAN`, and the corresponding boolean value as token value. Print appropriate messages.

The following program should print true and false.

```
begin
print(TRUE);
print(FALSE);
end
```

3. New boolean operator (and)

Add a new binary operator, `AND`, producing the token `AND`. The syntax is `x AND y` where `x` and `y` are boolean expressions.

This operator computes the logical *and*. This is, `TRUE AND TRUE = true`, `FALSE AND TRUE = false`, `TRUE AND FALSE = false` and `FALSE AND FALSE = false`.

Force left to right evaluation order. Print appropriate messages.

The following program should print true and false.

```
begin
print(1 < 2 AND TRUE);
print(FALSE AND 2 > 1 AND TRUE);
end
```